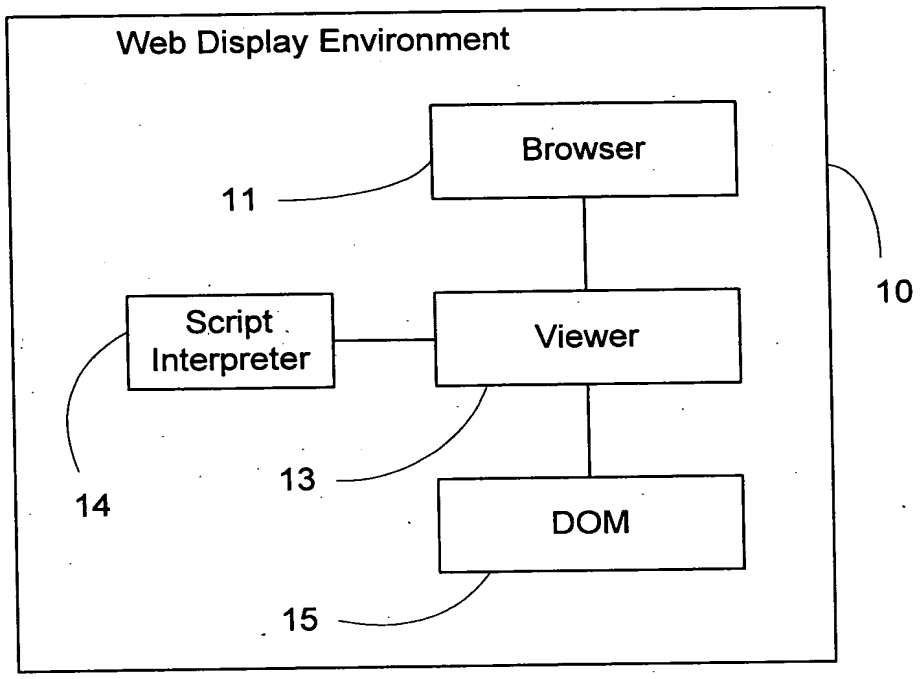


Figure 1
PRIOR ART

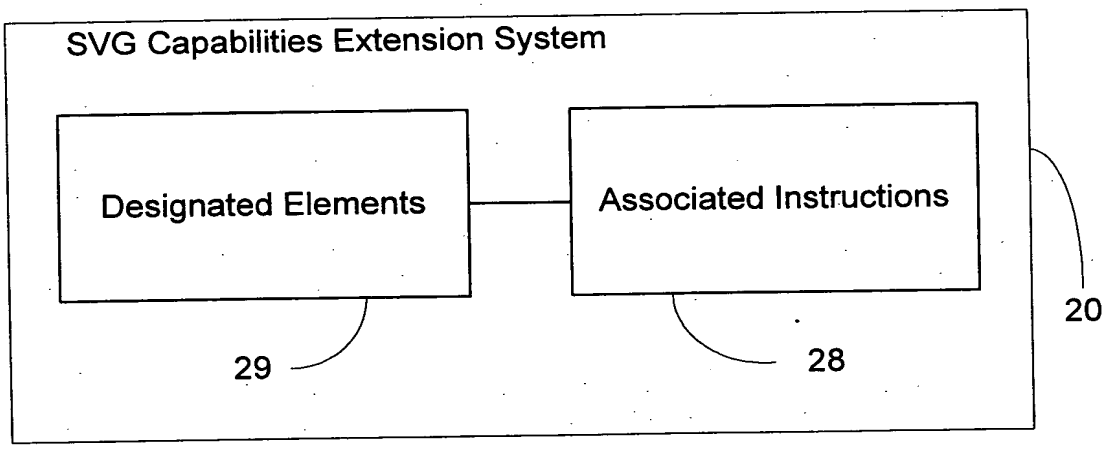


Figure 2

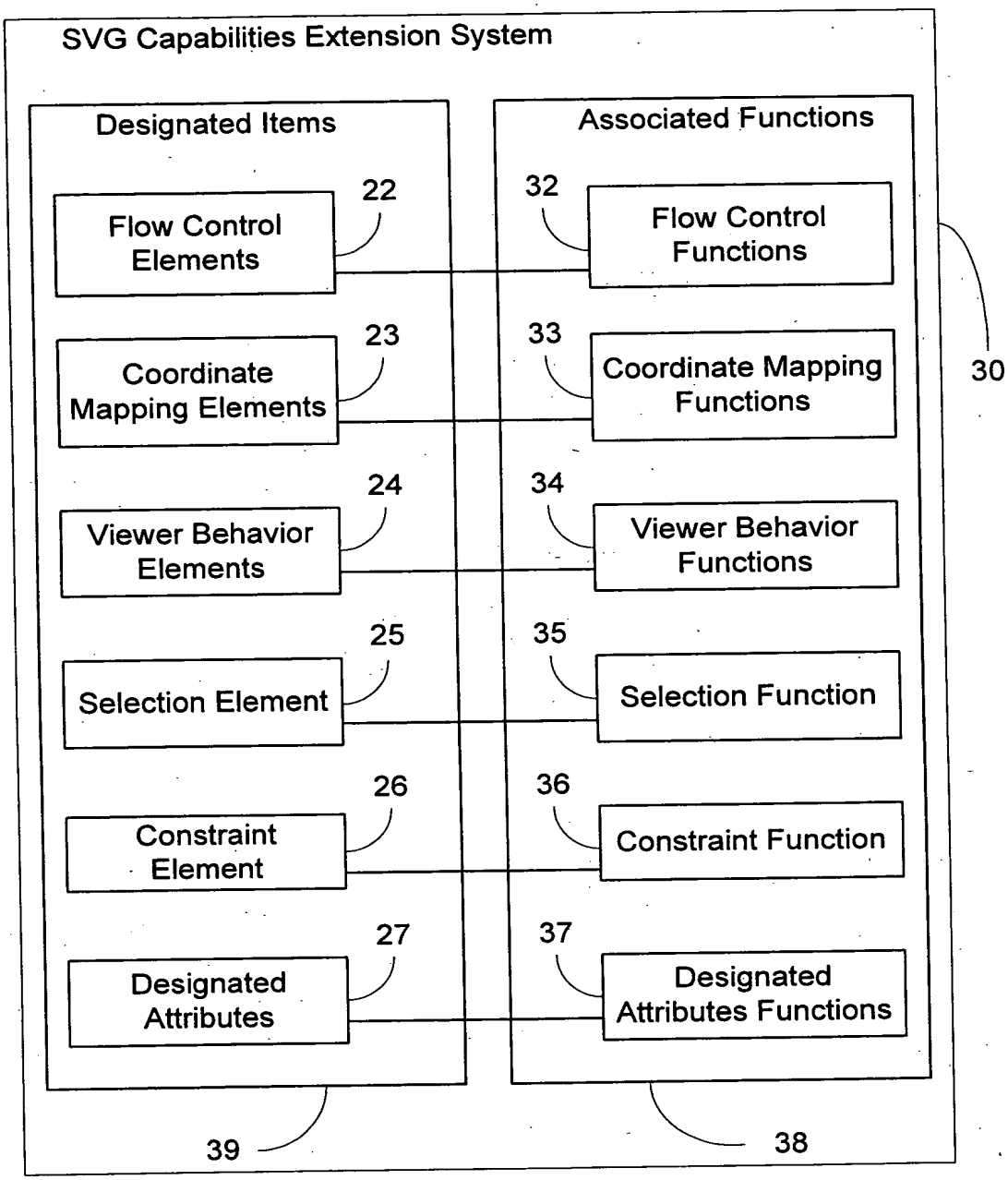

SVG Capabilities Extension System
Designated Items
Associated Functions
Flow Control Elements
22
Flow Control Functions
32
Coordinate Mapping Elements
23
Coordinate Mapping Functions
33
Viewer Behavior Elements
24
Viewer Behavior Functions
34
Selection Element
25
Selection Function
35
Constraint Element
26
Constraint Function
36
Designated Attributes
27
Designated Attributes Functions
37
30
39
38

Figure 3

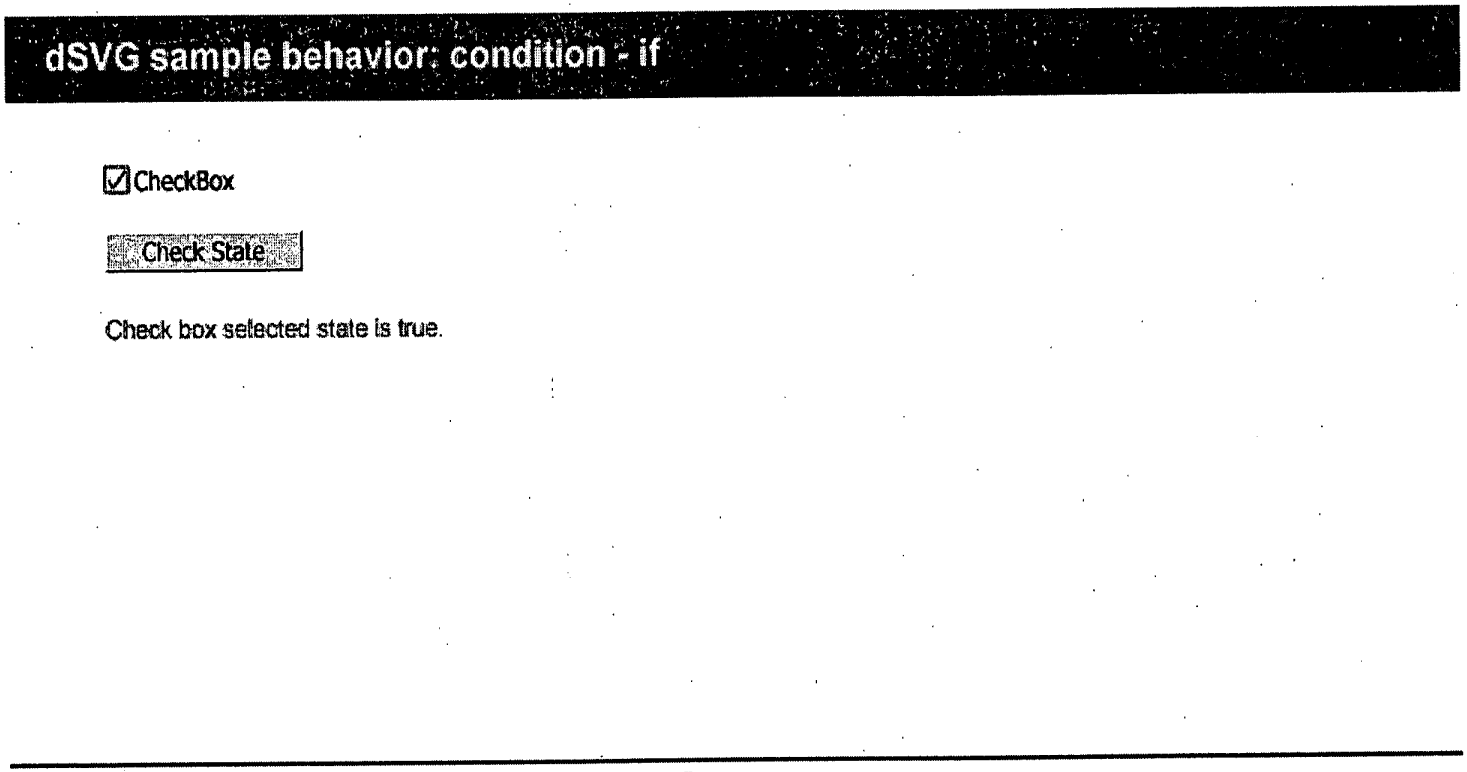

dSVG sample behavior: condition - if
CheckBox
Check State
Check box selected state is true.
Content of file: dsvg:checkBox, dsvg:Button, dsvg:if, dsvg:setData
The dsvg:if element executes or renders child elements based on a conditional if statement. (true/false)

Figure 4

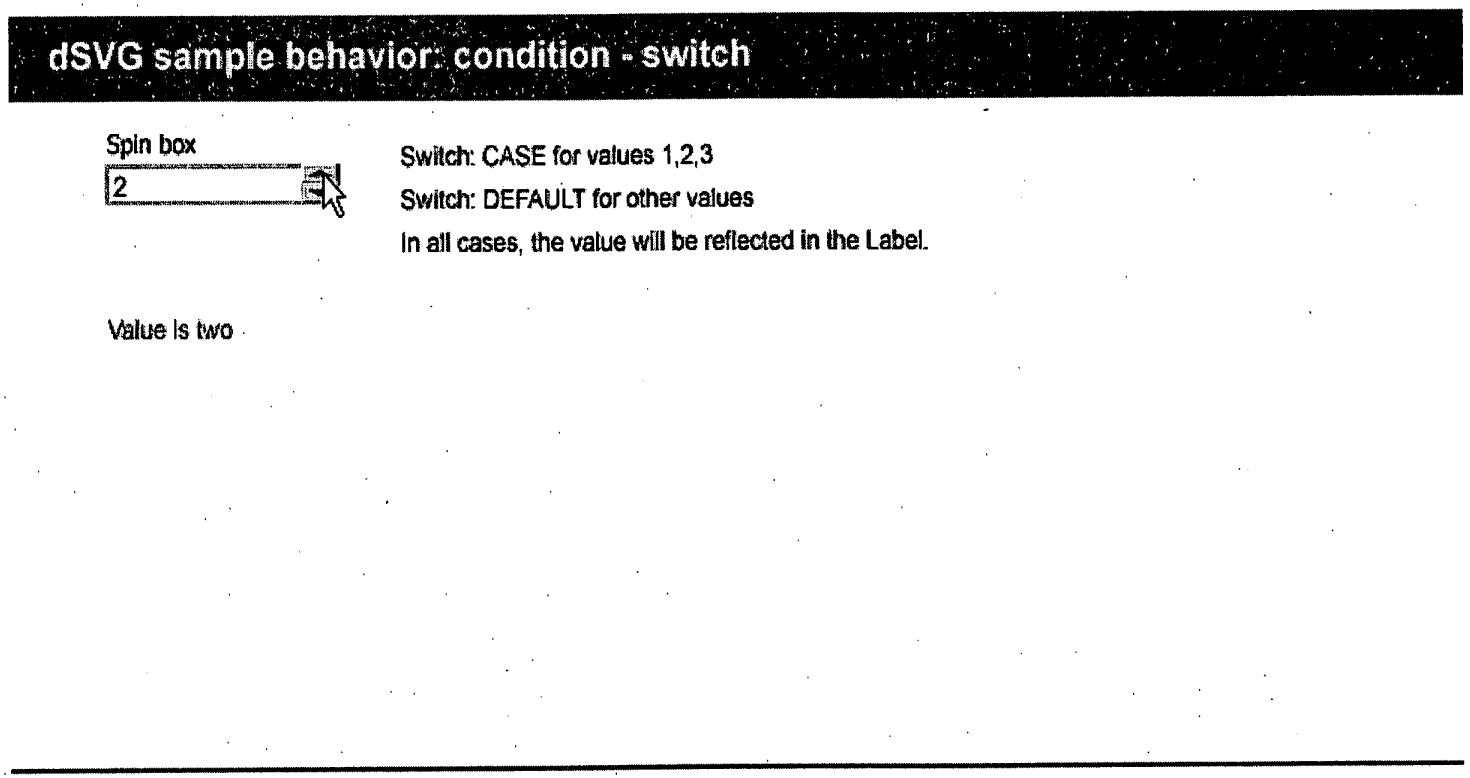

dSVG sample behavior: condition - switch
Spin box
2
Switch: CASE for values 1,2,3
Switch: DEFAULT for other values
In all cases, the value will be reflected in the Label.
Value is two
Content of file: dsvg:spinBox, dsvg:switch, dsvg:case, dsvg:default
The dsvg:switch element compares conditions of the child dsvg:case element(s) along with the dsvg:default element values.

Figure 5

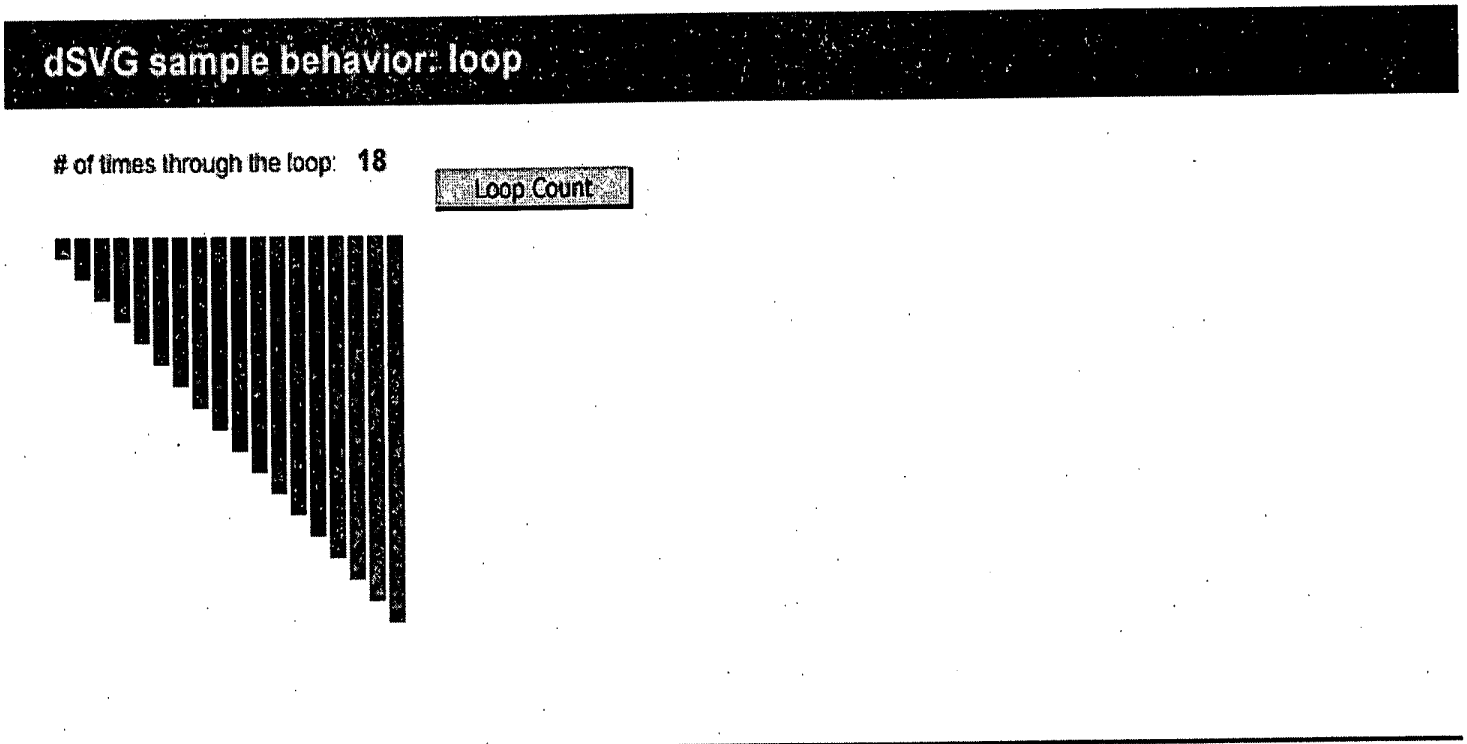

dSVG sample behavior: loop
# of times through the loop: 18
Loop Count
Content of file: dsvg:loop, dsvg:button, dsvg:setData, dsvg:setAttribute
The dsvg:loop element is a sequence of instructions that is continually repeated until a certain condition is reached.

Figure 6

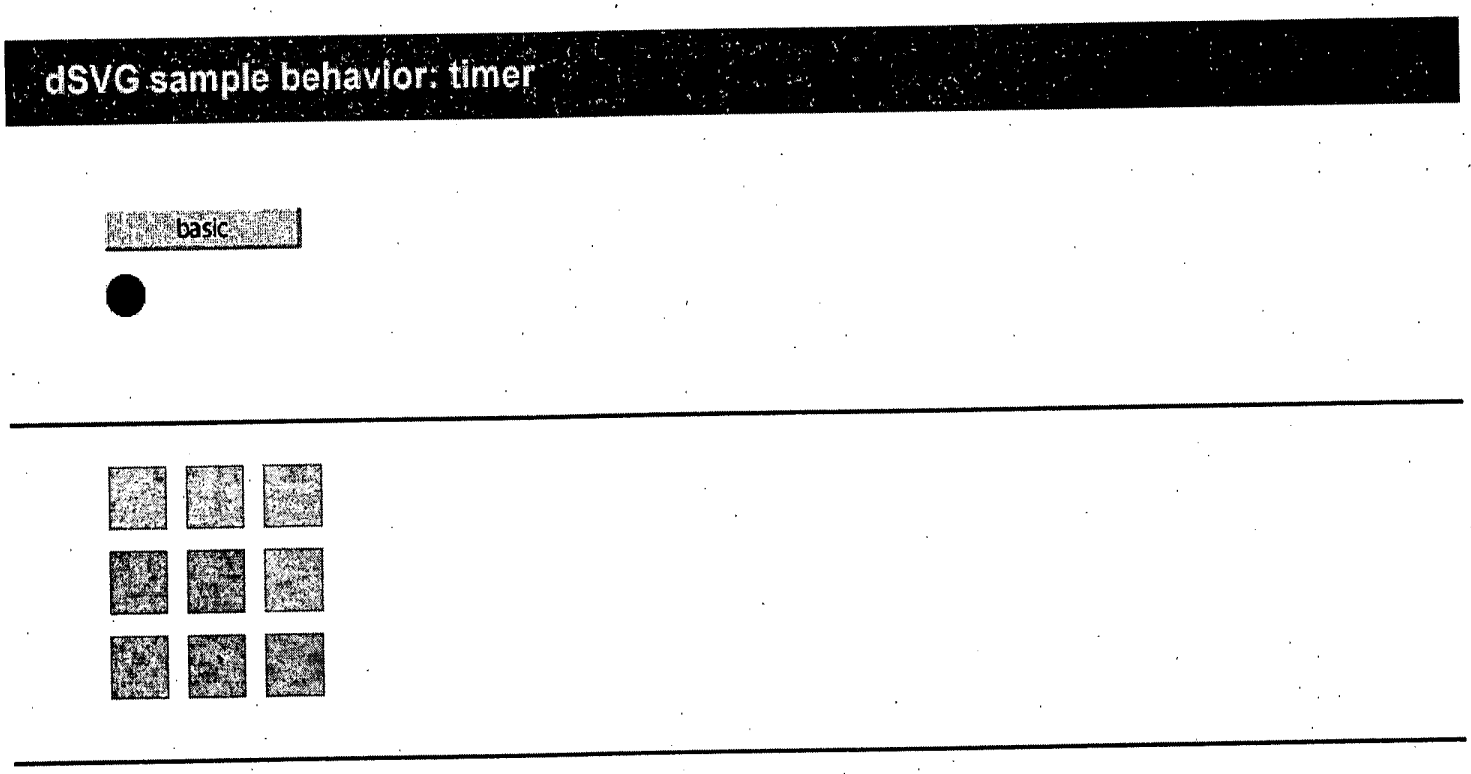

dSVG sample behavior: timer
basic
Selecting the button in the top portion will set the cx attribute for the circle.
The bottom set of rects has 2 timer applied.
1 moving forward to set each consecutive rect green. 1 starting at the last rect moving backwards turning each rect blue.

Figure 7

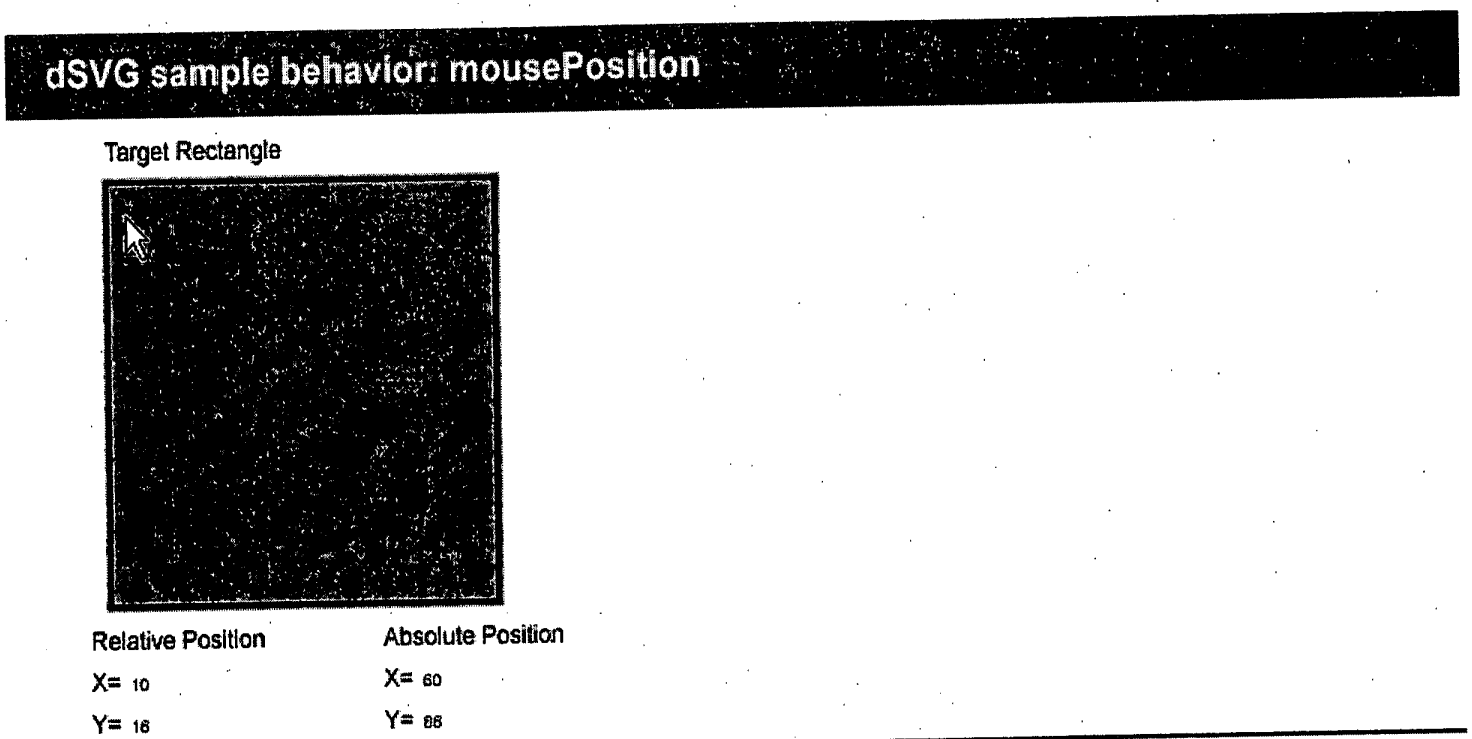
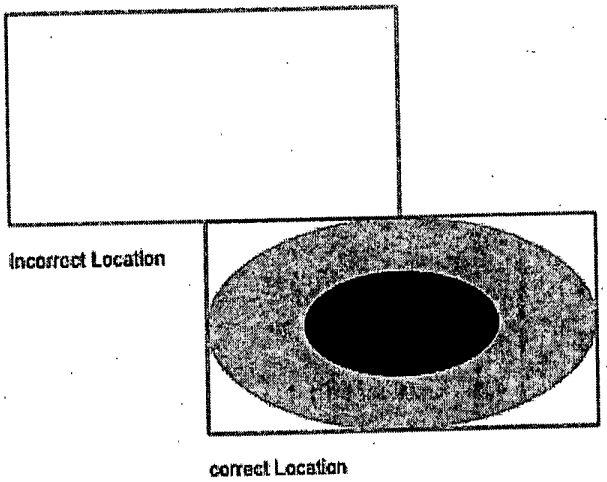

dSVG sample behavior: mousePosition
Target Rectangle
Relative Position
X= 10
Y= 16
Absolute Position
X= 60
Y= 86
Content of file: dsvg:mousePosition, dsvg:setData
The dsvg:mousePosition element defines a container for holding the current mouse coordinates.
The coordinates can be tracked relative to the document or absolute to the parent element.

Figure 8

dSVG sample behavior: mapCoords

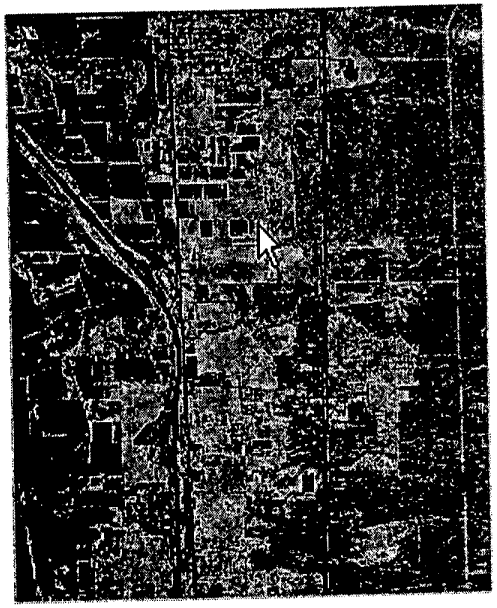


Content of file: dsvg:mapCoords, dsvg:pointPair, dsvg:setAttribute
The dsvg:mapCoords element defines an object used for mapping from one coordinate space to another.
The resulting coefficients are determined by the coordinates of the point-pairs (child) elements.

Figure 9

dSVG sample behavior: mapProj

Target Image



x,y: 126.5, 101

UTM: 348624, 3536350

Lat/Long: '31° 57' 10.92"N' '106° 36' 6.16"W'

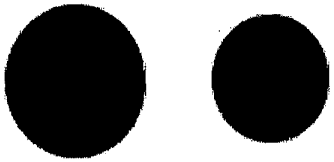
Content of file: dsvg:mapProj, dsvg:mapCoords, dsvg:pointPair, dsvg:setData, dsvg:mousePosition
The dsvg:mapProj element defines an object used for mapping coordinates from one project system to another.
For example, "latlong" can be mapped to "UTM".

Figure 10


dSVG sample behavior: alert
Fire alert
alert
Alert message
OK
Content of file: dsvg:button, dsvg:alert
The dsvg:alert element is a dialog box used to display a custom message.

Figure 11

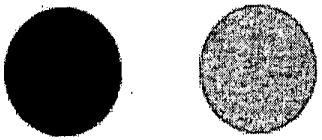
dSVG sample behavior: focus - with added attributes focusGroup and focus



red blue green orange

Click on the red, green and blue circles to set focus.

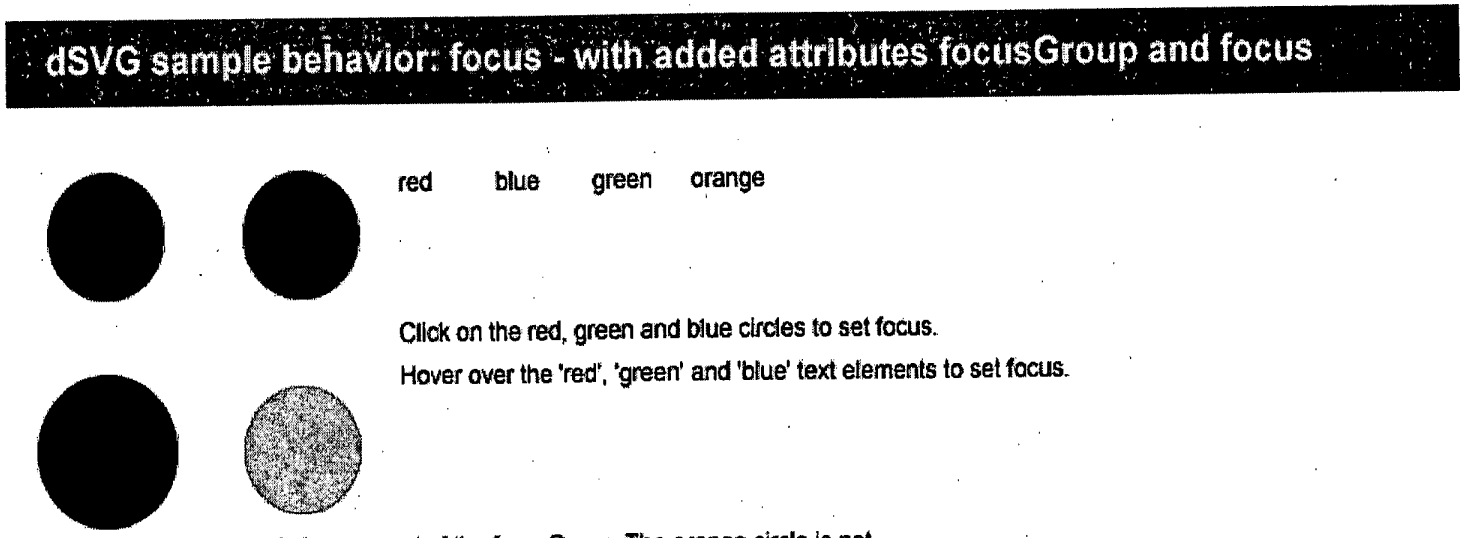
Hover over the 'red', 'green' and 'blue' text elements to set focus.



The red, blue, green circles are part of the focusGroup. The orange circle is not.

Content of file: dsvg:focus, dsvg:setTransform, dsvg:setAttribute, dsvg:setStyle, (added attributes dsvg:focus, dsvg:focusGroup)
The dsvg:focusGroup attribute adds the ability to store the ID of similar type elements that are assigned to that group.
Default focus can be given to an element (red circle above) by adding the dsvg:focus attribute to that element.

Figure 12A



Content of file: dsvg:focus, dsvg:setTransform, dsvg:setAttribute, dsvg:setStyle, (added attributes dsvg:focus, dsvg:focusGroup)
The dsvg:focusGroup attribute adds the ability to store the ID of similar type elements that are assigned to that group.
Default focus can be given to an element (red circle above) by adding the dsvg:focus attribute to that element.

Figure 12B

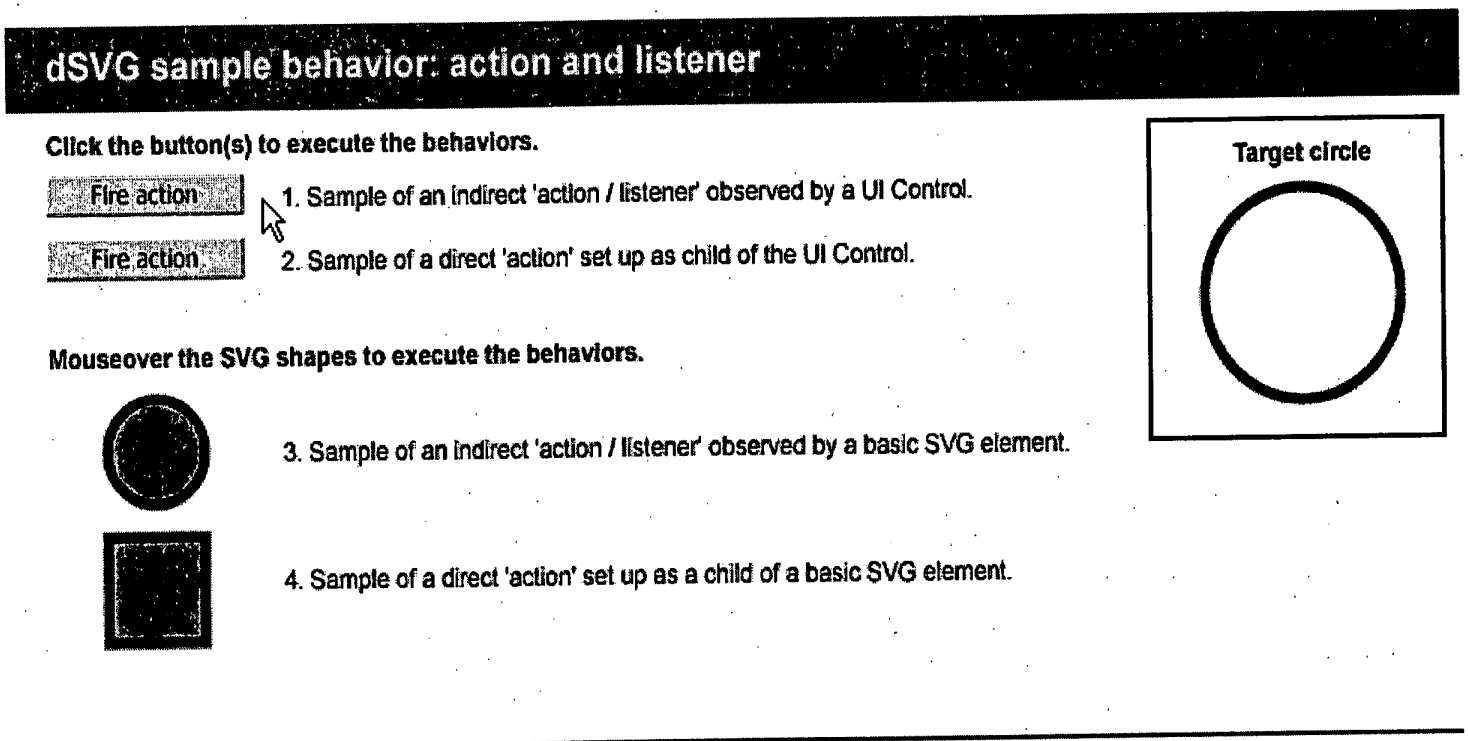

dSVG sample behavior: action and listener
Click the button(s) to execute the behaviors.
Fire action
1. Sample of an indirect 'action / listener' observed by a UI Control.
Fire action
2. Sample of a direct 'action' set up as child of the UI Control.
Target circle
Mouseover the SVG shapes to execute the behaviors.
3. Sample of an indirect 'action / listener' observed by a basic SVG element.
4. Sample of a direct 'action' set up as a child of a basic SVG element.
Content of file: dsvg:action, dsvg:listener
The dsvg:action element is a container for other dSVG behavior elements.
Actions can be associated indirectly using a listener element, or they can be set up directly as a child of an observing element.

Figure 13A

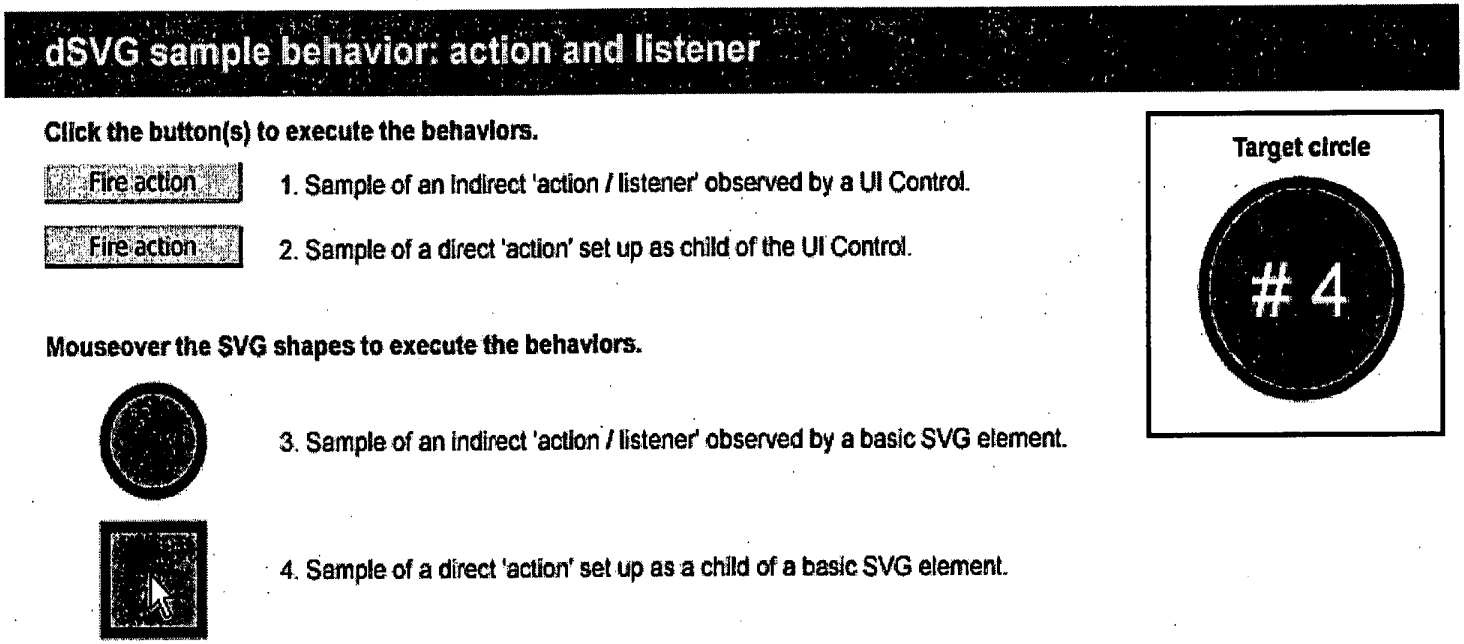

dSVG sample behavior: action and listener
Click the button(s) to execute the behaviors.
Fire action
1. Sample of an indirect 'action / listener' observed by a UI Control.
Fire action
2. Sample of a direct 'action' set up as child of the UI Control.
Target circle
4
Mouseover the SVG shapes to execute the behaviors.
3. Sample of an indirect 'action / listener' observed by a basic SVG element.
4. Sample of a direct 'action' set up as a child of a basic SVG element.
Content of file: dsvg:action, dsvg:listener
The dsvg:action element is a container for other dSVG behavior elements.
Actions can be associated indirectly using a listener element, or they can be set up directly as a child of an observing element.

Figure 13B

dSVG sample behavior: variable

width = 50

width = 50

Note: Once the button is selected, setAttribute is applied to the blue rect so width="previous '$varRect' value"

$varRect = redRect@width + blueRect@width

New Variable $varRect = 100

Content of file: dsvg:variable
The dsvg:variable element is able to assume different values.
Selecting the button will set a new value for the 'variable' ($varRect).

Figure 14A

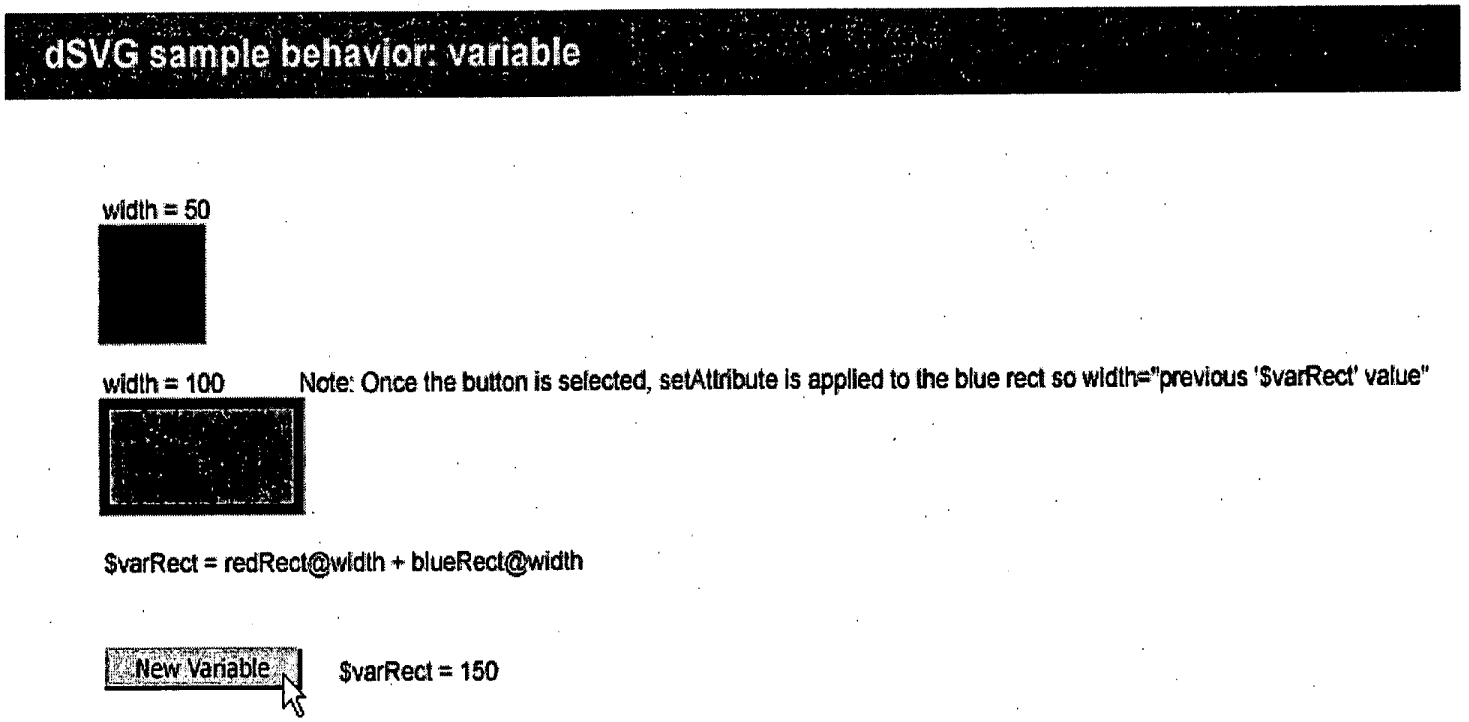

dSVG sample behavior: variable
width = 50
width = 100
Note: Once the button is selected, setAttribute is applied to the blue rect so width="previous '$varRect' value"
$varRect = redRect@width + blueRect@width
New Variable
$varRect = 150
Content of file: dsvg:variable
The dsvg:variable element is able to assume different values.
Selecting the button will set a new value for the 'variable' ($varRect).

Figure 14B

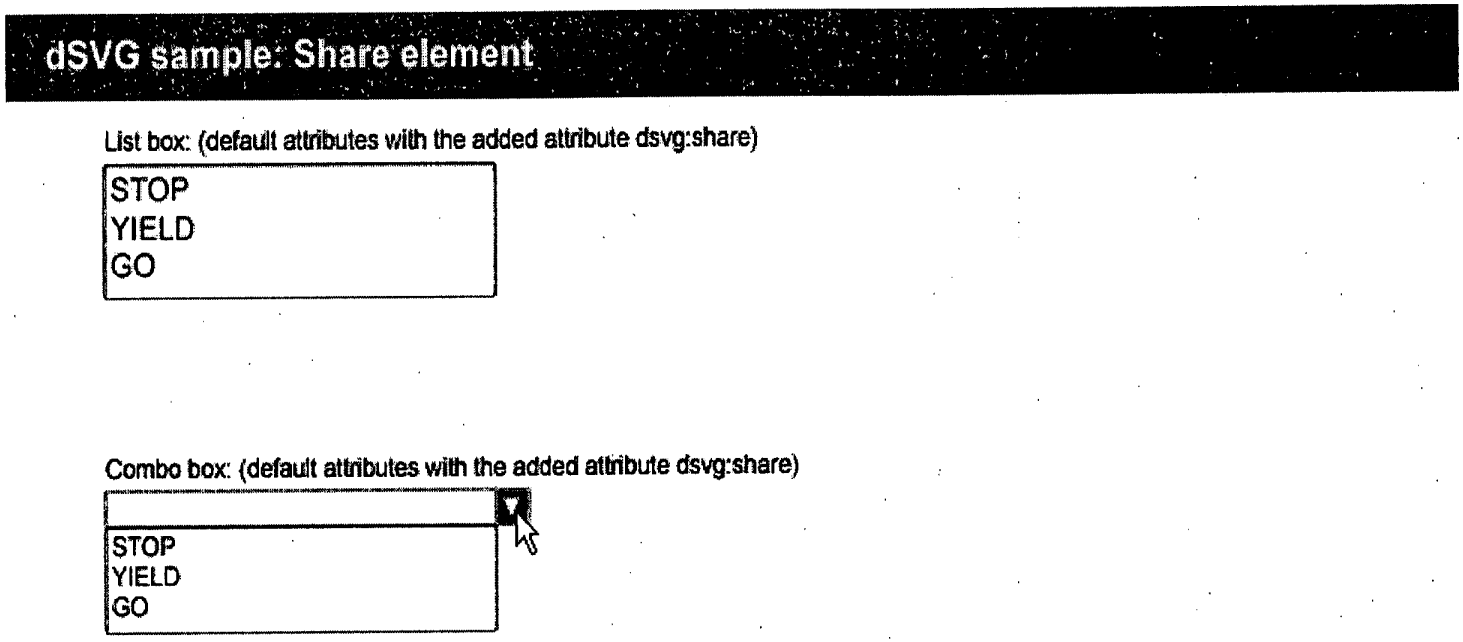

dSVG sample: Share element
List box: (default attributes with the added attribute dsvg:share)
STOP
YIELD
GO
Combo box: (default attributes with the added attribute dsvg:share)
STOP
YIELD
GO
The share element is used to share a group of items with multiple elements.
This document shares the same set of items with the combo box and the list box.
Associate a share element with other elements by adding a dsvg:share attribute to the element that references the share element.

Figure 15

dSVG sample: drag (added attribute)

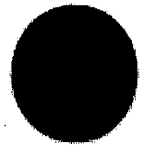
Select each of the objects and attempt to drag to another position.

Blue circle has drag="true"

Button has drag="true"

drag (true)

Red circle has drag="false"



Button has drag="false"

drag (false)

Content of file: dsvg:drag
The dsvg:drag attribute is applied to elements to set the drag to either true or false.

Figure 16A

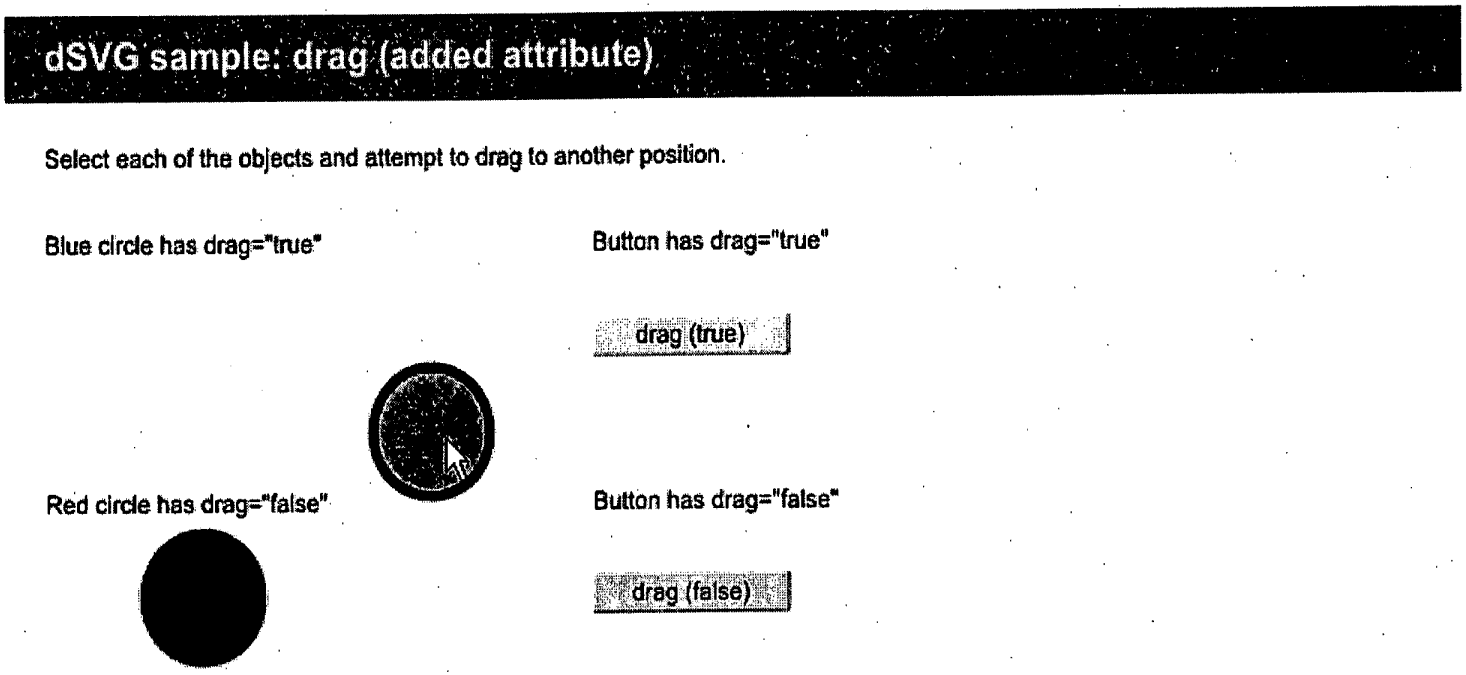

dSVG sample: drag (added attribute)
Select each of the objects and attempt to drag to another position.
Blue circle has drag="true"
Button has drag="true"
drag (true)
Red circle has drag="false"
Button has drag="false"
drag (false)
Content of file: dsvg:drag
The dsvg:drag attribute is applied to elements to set the drag to either true or false.

Figure 16B

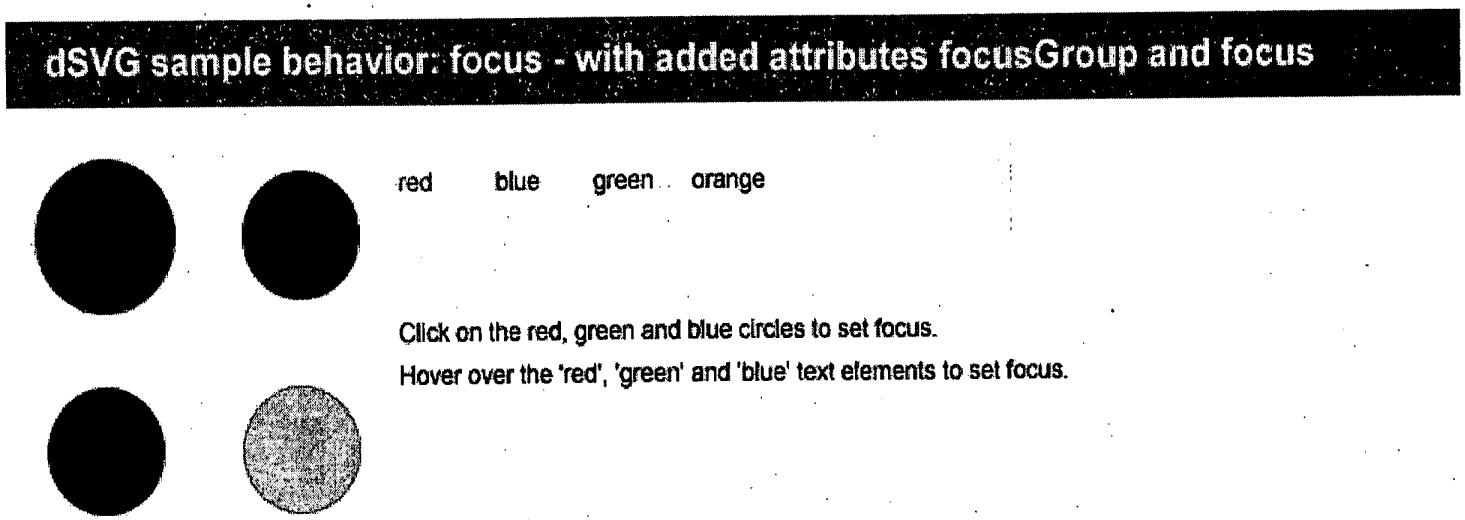
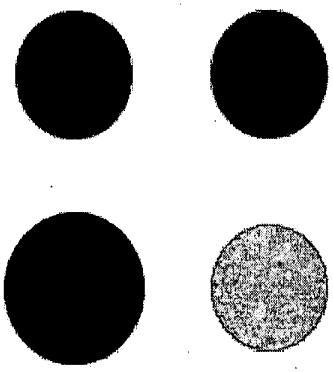

dSVG sample behavior: focus - with added attributes focusGroup and focus
red
blue
green
orange
Click on the red, green and blue circles to set focus.
Hover over the 'red', 'green' and 'blue' text elements to set focus.
The red, blue, green circles are part of the focusGroup. The orange circle is not.
Content of file: dsvg:focus, dsvg:setTransform, dsvg:setAttribute, dsvg:setStyle, (added attributes dsvg:focus, dsvg:focusGroup)
The dsvg:focusGroup attribute adds the ability to store the ID of similar type elements that are assigned to that group.
Default focus can be given to an element (red circle above) by adding the dsvg:focus attribute to that element.

Figure 17A

dSVG sample behavior: focus - with added attributes focusGroup and focus



red blue green orange

Click on the red, green and blue circles to set focus.

Hover over the 'red', 'green' and 'blue' text elements to set focus.

The red, blue, green circles are part of the focusGroup. The orange circle is not.

Content of file: dsvg:focus, dsvg:setTransform, dsvg:setAttribute, dsvg:setStyle, (added attributes dsvg:focus, dsvg:focusGroup)
The dsvg:focusGroup attribute adds the ability to store the ID of similar type elements that are assigned to that group.
Default focus can be given to an element (red circle above) by adding the dsvg:focus attribute to that element.

Figure 17B

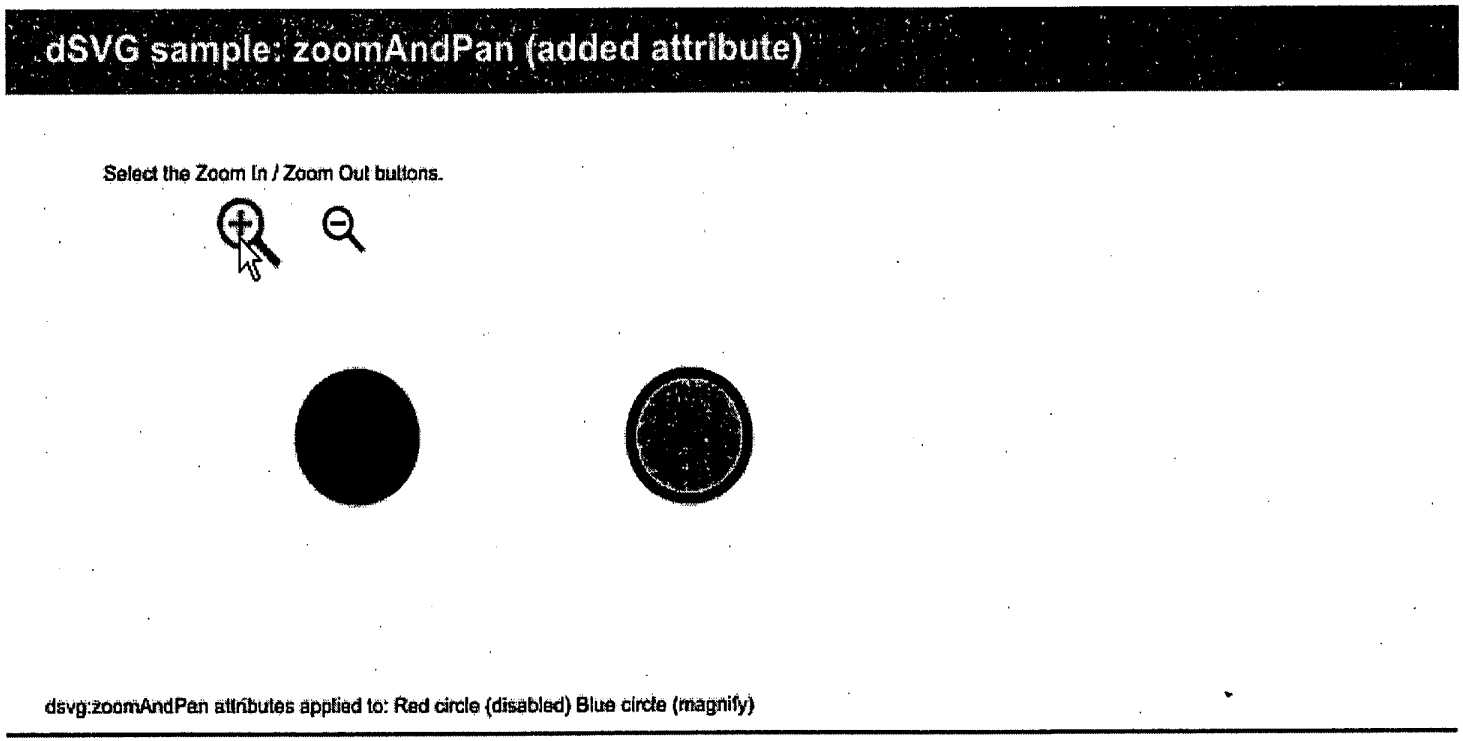

dSVG sample: zoomAndPan (added attribute)
Select the Zoom In / Zoom Out buttons.
dsvg:zoomAndPan attributes applied to: Red circle (disabled) Blue circle (magnify)
Content of file: dsvg:zoom, dsvg:zoomAndPan
The dsvg:zoom element will zoom in / zoom out by the amount specified in the scale attribute.

Figure 18A

Select the Zoom In / Zoom Out buttons.

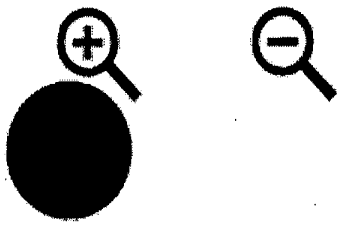


Figure 18B

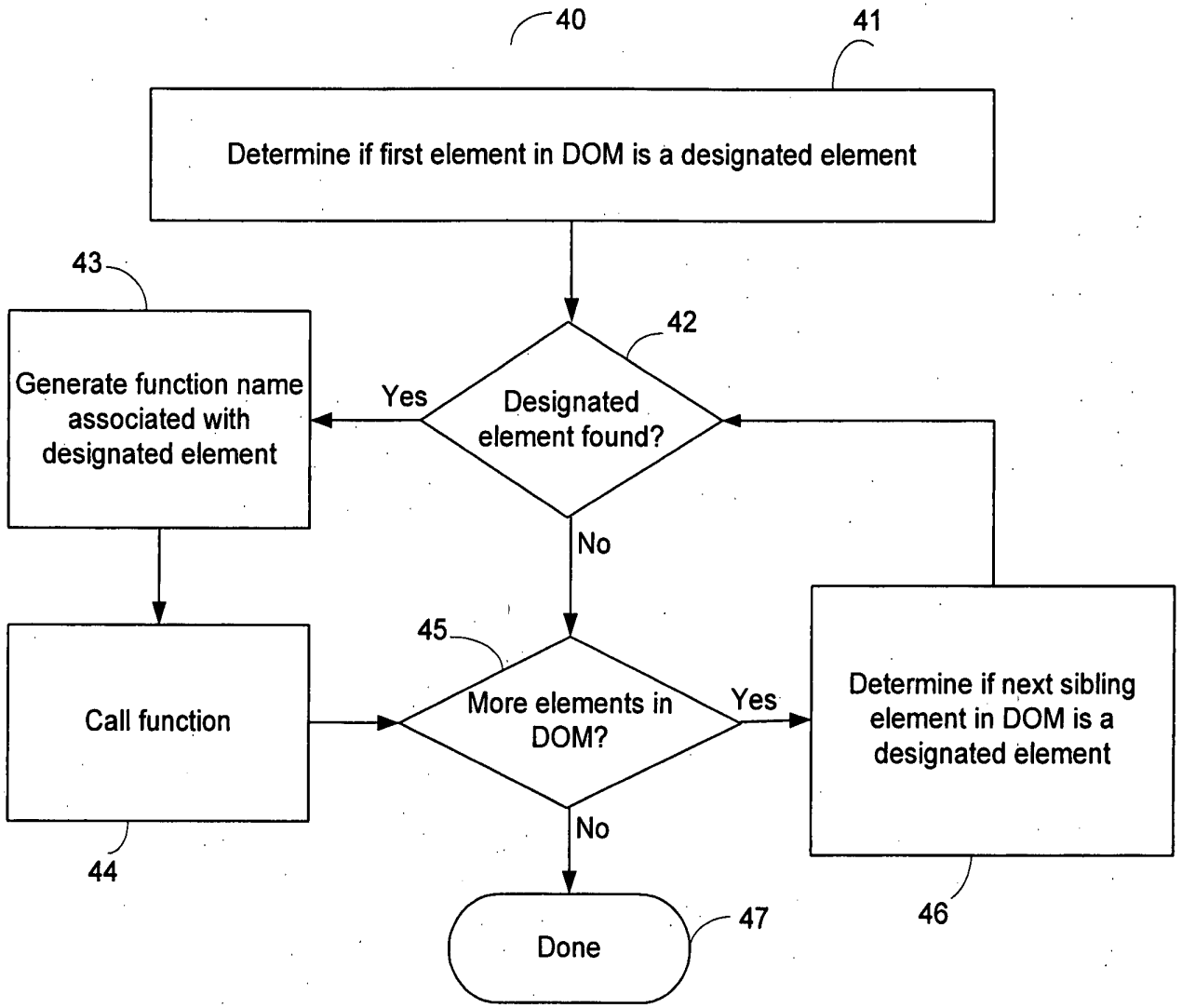

40
41
Determine if first element in DOM is a designated element
43
42
Generate function name associated with designated element
Yes
Designated element found?
No
45
Call function
More elements in DOM?
Yes
Determine if next sibling element in DOM is a designated element
No
44
46
Done
47

Figure 19

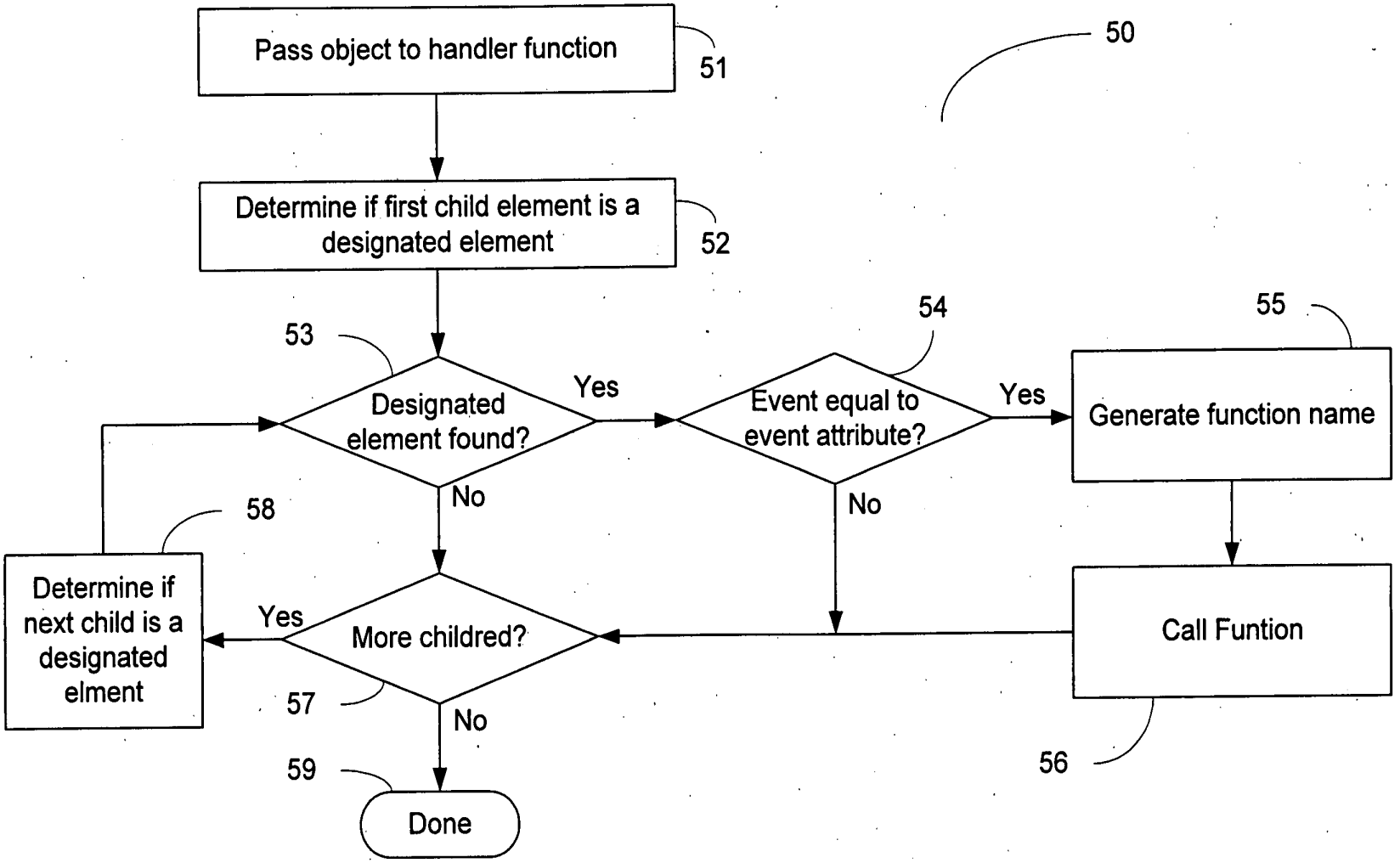

Pass object to handler function
51
50
Determine if first child element is a designated element
52
53
Designated element found?
Yes
54
Event equal to event attribute?
Yes
55
Generate function name
No
No
58
Determine if next child is a designated elment
Yes
More childred?
Call Funtion
57
No
56
59
Done

Figure 20

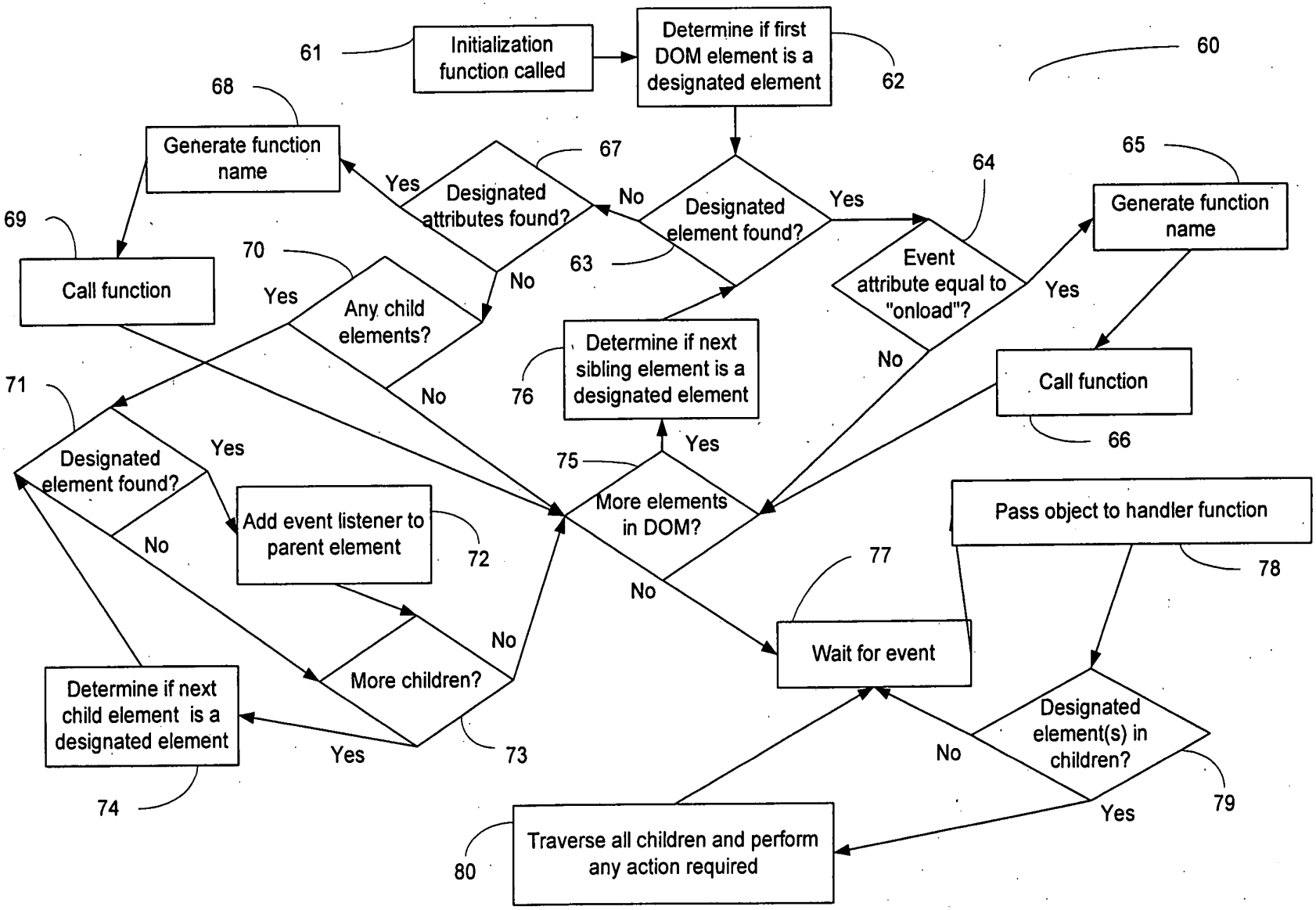

61
Initialization function called
Determine if first DOM element is a designated element
62
60
68
Generate function name
67
Designated attributes found?
Yes
No
Designated element found?
Yes
64
65
Generate function name
69
Call function
70
Yes
Any child elements?
No
63
Event attribute equal to "onload"?
Yes
No
71
Designated element found?
Yes
No
Add event listener to parent element
72
76
Determine if next sibling element is a designated element
Call function
66
75
Yes
More elements in DOM?
No
Pass object to handler function
78
77
Wait for event
No
More children?
Yes
73
Determine if next child element is a designated element
74
Designated element(s) in children?
No
Yes
79
80
Traverse all children and perform any action required

Figure 21